IN THE CLAIMS

The claims are amended as follows:

Claims 1 to 9 (canceled)

10. (currently amended) The system of ~~claim 9~~ claim 13, wherein the each data container in the plurality of data containers returns unused storage to the storage pool.

11. (canceled)

12. (canceled)

13. (currently amended) ~~The system of claim 12, wherein~~ A system for dynamic file allocation, comprising:

an input device for receiving a data object;

a reference container, coupled to the input device, for directing the data object to a location in a storage object, coupled to the reference container, the reference container comprising a publishing point which contains a virtual mapping system that directs the data object to an entry in the storage object;

the storage object comprising a plurality of data containers;

the virtual mapping system comprising a reference object which is assigned to the data object and comprises a set of bits defining the location of the data object in the plurality of data containers where the data object is located;

each data container in the plurality of data containers comprising a transfer space to temporarily store data objects being relocated, a data space, coupled to the transfer space, to store the data object, and an available space, coupled to the data space, containing unused space;

the publishing point comprising an import space for temporarily storing the data object, a reference space, coupled to the import space, for storing the reference objects, and an available space, coupled to the reference space, containing unused storage in

the publishing point;

the data object being selected from a group consisting of a byte, a file, and a directory;

each data container in the plurality of data containers obtaining storage from a storage pool, wherein the size of the storage pool can be increased by adding more storage, and wherein power is not disrupted in the file system and wherein an operator sets a maximum size to each data container in the plurality of data containers; and

the operator ~~sets~~ setting a relocation threshold on the each data container in the plurality of data containers, which indicates when the data object is to be relocated from a first data container in the plurality of data containers to a second data container in the plurality of data containers.

14. (original) The system of claim 13, wherein the relocation threshold is a percentage of the maximum allowable size of the each data container in the plurality of data containers.

15. (canceled)

16. (currently amended) The system of claim 13, wherein the data object to be relocated from the first data container ~~to be relocated~~ is placed in the transfer space of the second data container.

17. (original) The system of claim 16, wherein the data object is relocated from the first data container to the second data container without the knowledge of an application running on the system.

18. (currently amended) ~~The system of claim 2, wherein~~ A system for dynamic file allocation, comprising:

an input device for receiving a data object;

a reference container, coupled to the input device, for directing the data object to a location in a storage object, coupled to the reference container, the reference

container comprising a publishing point which contains a virtual mapping system that directs the data object to an entry in the storage object;

the storage object comprising a plurality of data containers;

each data container in the plurality of data containers ~~is~~ being coupled to a container monitor to monitor that the each data container in the plurality of data containers has adequate available space.

19. (original) The system of claim 18, wherein a mode manager, coupled to the publishing point, maintains all internal data structures tracking the publishing point, the plurality of data containers and the data object.

20. (original) The system of claim 19, wherein a relocator is spawned by the mode manager to assist with the relocation of the data object.

Claims 21 to 25 (canceled)

26. (currently amended) ~~The method of claim 25, further comprising the steps of:~~ A method of dynamically allocating a data object in a file system, the method comprising the steps of:

detecting a data object to be relocated in a data space of a source data container;

copying the data object to the transfer space of a target data container;

transferring the data object from the transfer space of the target data container to the data space of the target data container;

updating a reference object to indicate the address of the data object in the target data container;

storing the reference object in a reference space in a publishing point;

spawning a relocator to manage the transfer of the data object from the source data container to the target data container;

the relocator updating the reference objects;

verifying the copy of the data object in the source data container is identical to the data object in the target data container;

moving the data object in the data space in the source data container to the transfer space in the source data container; and

removing the data object from the transfer space of the source data container to increase available space in the source data container for additional data objects.

27. (original) The method of claim 26, wherein the relocator moves the data object in the source data container to the transfer space and then removes the data object from the source data container.

28. (original) The method of claim 27, wherein a mode manager cycles through the source data container and the target data container to test for a relocation threshold.

29. (original) The method of claim 28, wherein an operator sets a time interval to test the relocation threshold.

30. (original) The method of claim 28, further comprising the steps of building a list of data objects from the source data container to relocate to the target data container, wherein data objects are put onto the list if the source data container has reached its relocation threshold.

31. (original) The method of claim 30, wherein the data object that is relocated is of sufficient size to cause the source data container to drop below its relocation threshold.

32. (original) The method of claim 31, further comprising the step of increasing the size of the source data container by obtaining additional space from a storage pool.

33. (original) The method of claim 32, wherein the mode manager determines where to relocate the data object and spawns the relocator to assist in the relocation.

34. (original) The method of claim 33, wherein the mode manager polls the import space of the publishing point for the data object to be relocated and wherein the operator sets a time interval for polling the import space.

35. (original) The method of claim 33, wherein the data object is relocated to multiple data containers if the size of the data object is too large to fit into a single data container.

Claims 36 to 43 (canceled)

44. (currently amended) ~~The method of claim 43, wherein~~ A method of dynamically allocating a data object in a file system, the method comprising the steps of:

entering the data object into an input device;

storing the data object in an import space of a publishing point, wherein additional storage space is acquired from an available space of the publishing point if there is not enough space in the import space;

assigning a reference object to the data object;

relocating the data object to a transfer space of a data container chosen from a plurality of data containers;

the reference object indicating the address in the data container where the data object is stored;

the reference object being transferred from the import space of the publishing point to the reference space of the publishing point;

a mode manager spawning a relocator to assist in relocating the data object from the import space of the publishing point to the transfer space of the data container;

the mode manager cycling through each container in the plurality of data containers to test for a relocation threshold; and

an operator ~~sets~~ setting a time interval to test the relocation threshold.

45. (original) The method of claim 44, further comprising the steps of:

moving the data object from the transfer space of the data container to the data space of the data container; and

updating the reference object with the address of the data object in the data container.

46. (original) The method of claim 45, wherein the data object is relocated to multiple data containers if the size of the data object is too large to fit into a single container.

47. (original) The method of claim 46, wherein the mode manager polls the import space of the publishing point for the data object to be relocated and wherein the operator sets a time for polling the import space.

48. (new) The method of claim 44, wherein the mode manager spawns a relocator for each data object in the plurality of data containers.